On the Performance of Simple Parallel Computer of Four PCs Cluster

H. K. Dipojono and H. Zulhaidi

High Performance Computing Laboratory
Department of Engineering Physics
Institute of Technology Bandung
Jln. Ganesha 10 Bandung 40132
E-mail : dipojono@tf.itb.ac.id

Abstract

We have constructed a simple parallel computer in the form of four Pentium III 600 MHz PCs cluster. The simplicity of our cluster was intentionally chosen so that others can follow with ease. We used Linux and MPI for the operating system and clustering application, respectively. Benchmarkings were conducted by using some public benchmark applications such as POVRAY and HPL (high performance Linpack). In addition we have also tested the performance of our PCs cluster for some heavy matrix manipulations. Our cluster shows bandwidth utility up to 90% and power more than 3.8 times that of one PC. For povray benchmark our cluster shows an index of 1138.46, far beyond that of Cray T3E-900 AC64.

Key words: parallel computer, cluster, Linux, MPI

1. Introduction

The human need of computing power will follow with their curiosity toward nature. The more they want to explore their curiosity about nature the more they need that of computing power. Although the technology could provide their need of it nevertheless the price they have to pay is increasingly much faster than they can afford it. The price of supercomputer is non-linearly increasing with its power. The consequence is quite obvious, the number of universities or research institutions that can afford it are decreasing. Fortunately the technology could provide an alternative through PCs clustering, an approach first popularized by the Beowulf project [1].

PCs cluster built using hardware components dan free software are only recently received wide attention from both research and education institutions. It emergence is a consequence result of the convergence of several trends, including but not limited to the availability of inexpensive high performance microprocessors and high speed networks, the development of tools for high performance distributed computing, and the endlessly increasing need of computer power from both scientific and commersial communities. More importantly, cluster computing provides an inexpensive computing resource to educational institutions. Further details on this clustering could be found in [2].

We have constructed a simple parallel computer in the form of four Pentium III 600 MHz PCs cluster. Linux [3] and MPI [4,5,6] are used for the operating system and clustering application, respectively. This paper will present our experience working on simple cluster computing with the hope that the more complex cluster computing is just a matter of expansion. We will show the performance of our cluster based on benchmarks varied from the low-level single CPU and subsystem tests including parallel communication to the simple real parallel application tests.

2. Cluster Environment

The Linux cluster which was used here consisted of four PCs and the systems were attached together with a fast Ethernet using switch hub of 16 ports. The node 1, 2 and 3 used processor of Pentium III 600EB MHz frequency with the amount of memory 64 MB and disk of

20 GB. The node 4 used processor of AMD Athlon 850 MHz with the amount of memory 128 MB. The systems were running Linux Mandrake 8.1 with a kernel level of 2.4.17. We used public domain gnu compilers (C, C++, Fortran). The MPI implementation used MPICH [7] 1.2.2.3 from Argonne National Laboratory with a standard TCP/IP communication layer.

The cluster had its own private network and nevertheless the server was connected both to the public local area network and the private cluster network. The design was such that users could use the cluster from the remote machines. The communication among the nodes was based on switch of 100 Mbits/second. The switch hub had 16 ports from which 3 ports were used for the computing nodes and one for the server. Star topology was used in our cluster and Figure 1 shows our network topology.

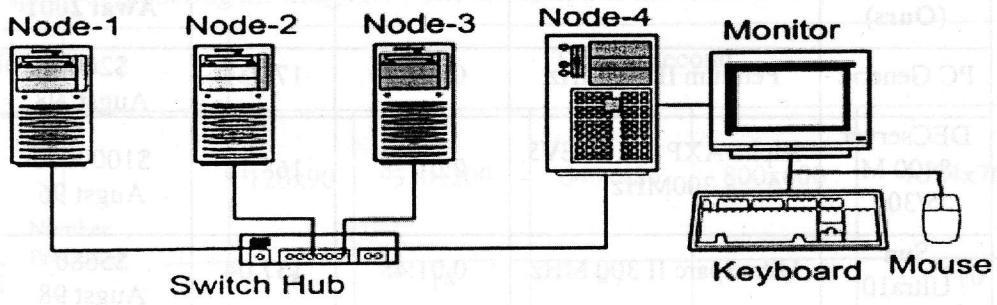

Figure 1 The network topology of our cluster

Mosix of a kernel 2.4.17 was used for cluster management. For job monitoring of each node, we installed mosixview [8]. These applications are freely available. Basically, the Mosix system provides tools for tranparent process migration. As already well known as much our experience showed, the most time consuming in building the cluster environment were getting the programming environment to work as well as tuning the communication network. Building the hardware obviously was the simplest one.

3. Performance

To show that our system worked we had benchmarked the system from the low-level single node and subsystem tests including communication network to the real parallel application tests. In the applications some free reference tests, such as Povray [9], were used. By using these applications comparison of the performance between single processor and that of four processors were conducted. We used a modified Povray, an open source code for image processing applications, to test our system. Modification were made in order to match with the requirements of MPI cluster computation. The parameter used in our test was a standard, an image of a vase, given by the source code. This standard provides the results of the previous benchmarks. Figure 2 shows the standard image provided by the Povray source code. Table 1 shows the previous benchmarks by using the standard image of Figure 2.

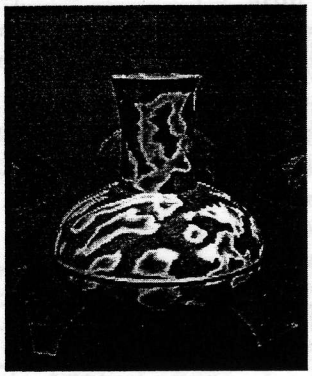

Figure 2. The standard image given by the source code

The reference given by Povray was that the Intel Pro of 200 MHz was used with computing time of 148 seconds and the index of povray at 100.

Table 1. Benchmarking for single processor

Results of benchmarking for single processor				
Machine	**Processor (speed)**	**Processing time**	**Povray Index**	**Price (Year)**
PC Generik (Ours)	**Pentium III 600MHz**	**0:00:47**	**314.89**	**$400 Awal 2001**
PC Generik	Pentium II 400MHz	0:01:23	178.31	$2000 Augst 98
DECserver 8400 M 5/300	Alpha AXP 21164 EV5 300MHz	0:01:28	168.18	$1000000 Augst 96
Sun Ultra10	UltraSparc II 300 MHz	0:01:48	137.04	$5680 Augst 98
Macintosh PowerPC	PowerPC 604 120 MHz	0:05:02	49.01	$4600 Febr 96

The results of benchmarking for multiprocessors are shown in Table 2. Our cluster worked very well with processing time of 13 seconds which was more than 3 times faster as that of one processor.

Table 2 Benchmarking for multiprocessors with ours in the first row.

Results of benchmarking for multiprocessors					
No	**Machine**	**Processor (speed)**	**Processing time**	**Povray Index**	**Price (Year)**
1*	**PC *cluster* with 4 machines**	**3 Pentium III 600MHz and 1 Athlon 850MHz**	**0:00:13**	**1138.46**	**?**
2	Cray T3E-900-AC64	Dec Alpha EV5 (48CPU) 450MHz	0:00:03	4933.33	$5500000 Nov 97
3	SunSparcstation (4processor) x 6 mesin	SuperSparc 51 MHz	0:00:30	493.33	$25000 x 6 mesin Jan 96
4	New College Linux Cluster	Pentium-2 300MHz x 2, Pentium-MMX 166 MHz x 3, Pentium 133MHz x 3, Pentium 90MHz x2	0:00:36	411.11	$10000 April 98
5	North Country	6 Pentium II 233MHz	0:00:32	462.50	$7000 April 98

Theoretically the speed of our cluster of four processors should be 4 times, rather than just 3.62, faster than that of just one processor. Communication delay between nodes and overhead of softwares were responsible for this result. By using an image of skyvase in Figure 2 we compared further on the speed of processing for the different resolutions. Table 3 shows the results for those different resolutions with the number of processors varied. Figure 3 shows time versus the number of processors for different resolution. We may conclude that the speed of processing with four processors is, on the average, about 3.8 times faster than that of just one processor.

Table 3. Time needed in rendering the image for different resolution and number of processors

The speed of rendering the image for different resolution and number of processors						
No		Time in second				
Resol. / Number Processor		120x90	320x200	640x480	800x600	1024x768
1		4	14	47	70	110
2	2	3	10	31	47	73
3	4	1	4	12	18	29

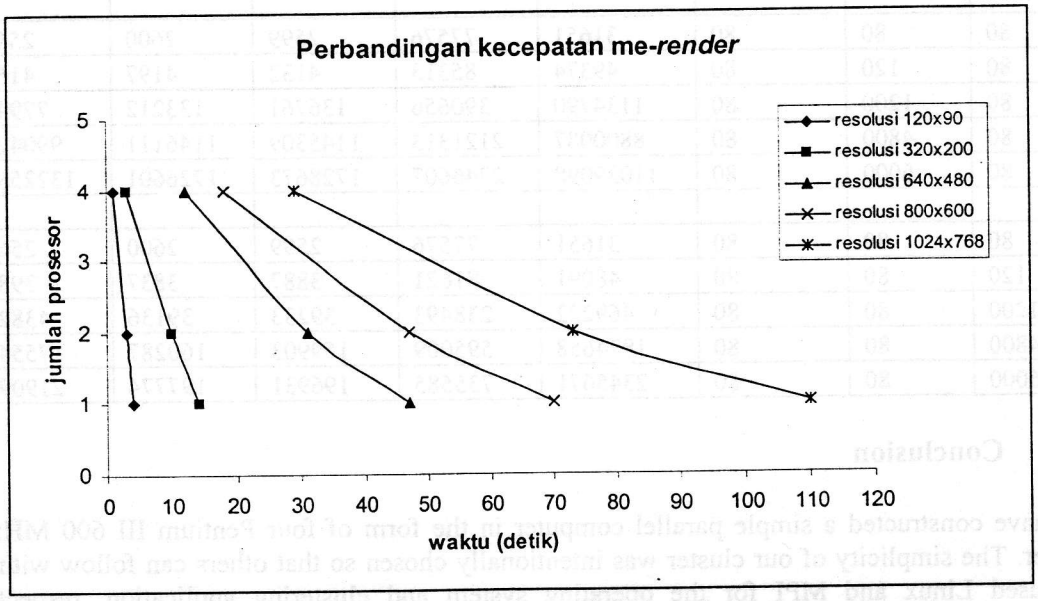

Figure 3 Time vs number of processors with the resolution as a parameter.

In order to gain more information on the performance of our cluster we also benchmarked on HPL [10] (high performance linpack) and matrix manipulation. Table 4 and 5 show our result and the reference, respectively.

Table 4 Gflops of our cluster for the problem given in the reference

GRID	1000	2000	3000	4000
1x4	0.458	0.773	0.899	1.066
2x2	0.509	0.764	0.894	1.04

Table 5 Gflops of the reference for the same problem in our benchmarked

GRID	2000	5000	8000	10000
1 x 4	1.28	1.73	1.89	1.95
2 x 2	1.17	1.68	1.88	1.93

For the matrix multiplications of two matrices, **A** and **B**, we defined NRA, NCA, and NCB as the number of rows of **A**, the number of column of **A**, and the number of column of **B**, respectively. The time of processing in parallel should be the sum of the processing time in each processor and the communication time. Table 6 shows the results for different matrix sizes. With the the communication time and software overhead dominated for small matrix sizes it is obvious that only for large matrix sizes clustering become an attractive alternative.

Table 6 Processing time for matrix multiplication

Input			Serial	Parallel			
NRA	NCA	NCB	Tser	Tpar	T1	T2	T3
80	80	80	31651	77576	2599	2600	2560
80	80	120	47223	79189	4864	4872	4116
80	80	1200	1089281	364993	136163	135509	70574
80	80	4800	4405474	1355563	591980	564996	591980
80	80	6000	5645864	1679305	716276	711715	340282
80	80	80	31651	77576	2599	2600	2560
80	120	80	49374	85313	4132	4197	4160
80	1200	80	1134790	390656	136761	133212	77944
80	4800	80	8800087	2121313	1145309	1146111	990424
80	6000	80	11039098	2746607	1728673	1726601	1322589
80	80	80	31651	77576	2599	2600	2560
120	80	80	48091	81621	3887	3837	3988
1200	80	80	469222	238493	39233	39136	43825
4800	80	80	1874658	595009	159903	160287	175543
6000	80	80	2345671	735585	196931	197774	219096

4. Conclusion

We have constructed a simple parallel computer in the form of four Pentium III 600 MHz PCs cluster. The simplicity of our cluster was intentionally chosen so that others can follow with ease. We used Linux and MPI for the operating system and clustering application, respectively. Benchmarkings were conducted by using some public benchmark applications such as POVRAY and HPL (high performance Linpack). In addition we have also tested the performance of our PCs cluster for some heavy matrix manipulations. Our cluster shows bandwidth utility up to 90% and power more than 3.8 times that of one PC. For povray benchmark our cluster shows an index of 1138.46, far beyond that of Cray T3E-900 AC64. Benchmarked on HPL shows that the performance of our cluster is below with that of the reference. The smaller of our cluster memory than that of the reference seems reponsible for the results. For the matrix multiplication our cluster shows that the cost on nodes communications and software overhead dominated the processing time for small matrix sizes. Hence it is only for large matrix sizes clustering will be an attractive alternative.

5. References

[1] Becker, Donald. Merkey, Phil. *The Beowulf project*. http://www.beowulf.org/

[2] Buyya, Rajkummar. 1999. *High Performance Cluster Computing*. New Jersey : Prentice-Hall,Inc.

[3] *Kernel Linux*. http://www.kernel.org/

[4] *LAM / MPI Parallel Computing. XMPI -- A Run/Debug GUI for MPI. http://www.lam-mpi.org/software/xmpi/*

[5] *MPIMap. http://www.llnl.gov/liv_comp/mpimap/*.

[6] *MPI supercomputer poster '96. http://www.netlib.org/mpi/*

[7] MPICH: http://www-unix.mcs.anl.gov/mpi/mpich/.

[8] *MOSIXVIEW Clustermanagment*. http://www.waplocater.de/mosixview/

[9] Povray image processing, http://www.povray.org/

[10] Dongarra, J. et al. HPL - *A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers*. http://www.netlib.org/benchmark/hpl/.